Leveraging Deep-learning and Field Experiment Response Heterogeneity to Enhance Customer Targeting Effectiveness

Kunpeng Zhang
University of Maryland, kzhang@rhsmith.umd.edu

Xueming Luo
Temple University, xueming.luo@temple.edu

Leveraging Deep-learning and Field Experiment Response Heterogeneity to Enhance Customer Targeting Effectiveness

Completed Research Paper

Kunpeng Zhang
University of Maryland
7699 Mowatt Ln, College park
kzhang@rhsmith.umd.edu

Xueming Luo
Temple University
1801 Liacouras Walk, Philadelphia
Xueming.Luo@temple.edu

Abstract

Firms seek to better understand heterogeneity in the customer response to marketing campaigns, which can boost customer targeting effectiveness. Motivated by the success of modern machine learning techniques, this paper presents a framework that leverages deep-learning algorithms and field experiment response heterogeneity to enhance customer targeting effectiveness. We recommend firms run a pilot randomized experiment and use the data to train various deep-learning models. By incorporating recurrent neural nets and deep perceptron nets, our optimal deep-learning model can capture both temporal and network effects in the purchase history, after addressing the common issues in most predictive models such as imbalanced training, data sparsity, temporality, and scalability. We then apply the learned optimal model to identify customer targets from the large amount of remaining customers with the highest predicted purchase probabilities. Our application with a large department store on a total of 2.8 million customers supports that optimal deep-learning models can identify higher-value customer targets and lead to better sales performance of marketing campaigns, compared to industry common practices of targeting by past purchase frequency or spending amount. We demonstrate that companies may achieve sub-optimal customer targeting not because they offer inferior campaign incentives, but because they leverage worse targeting rules and select low-value customer targets. The results inform managers that beyond gauging the causal impact of marketing interventions, data from field experiments can also be leveraged to identify high-value customer targets. Overall, deep-learning algorithms can be integrated with field experiment response heterogeneity to improve the effectiveness of targeted campaigns.

Keywords: Machine Learning, field experiments, deep-learning, customer targeting

Introduction

Most marketers recognize the value of customer targeting in their campaigns. Different customers may respond to the same marketing campaign (e.g., incentives) in significantly different ways. Identifying the right customers is critical to secure higher returns to investments in audience targeting (Agarwal et al. 2011; Anderson and Simester 2013; Ascarza 2018; Anderson and Simester 2013; Dube et al. 2017; Forbes 2015; Lambrecht and Tucker 2013; Lewis and Reiley 2014; Li et al. 2017; Liberali and Hauser 2018; Tucker 2014; Tucker and Zhang 2011; Yang and Ghose 2010). In common industry practices, companies generally identify customer targets based on their purchase history, i.e., frequent shoppers or high spenders. The primary challenge for marketers is how best to leverage heterogeneity in campaign responses in order to scientifically identify the proper customer targets.

Given such importance of customer targeting for marketers, prior research in both marketing and computer science has developed a variety of methods. Randomized experiments have been designed to allow researchers to gauge the causal impact of targeted campaigns and estimate heterogeneous treatment effects. A major challenge, however, is the cost of field experiments utilizing a company's whole customer base, as they tend to be expensive and occasionally harmful to firm performance. In addition, getting an organization's approval to conduct field experiments can be a long process, not to mention the cost of execution. Quantitative structural modeling may "predict" what happens when the world changes. It performs well with a small quantity of covariates or observed characteristics, but in the real world there are thousands of customer features to account for. In computer science, while a significant body of work develops machine-learning techniques to predict user purchase behavior and some use robust statistics (e.g. influence functions) to understand black-box predictions, they experience some common challenges in accounting for heterogeneity in the customer response to marketing interventions and capturing users' latent purchase patterns (Koh and Liang 2017).

To overcome these challenges, we propose to leverage deep-learning algorithms and field experiment response heterogeneity to enhance targeting effectiveness (see the overall framework in Figure 1). Specifically, in order to understand heterogeneity in customer responses to a marketing campaign, we recommend firms run a small-scale A/B test, i.e., a pilot randomized experiment with a small but representative sample of the company's customer base. Using the field experiment's data and observed customer characteristics, we build, train, and validate an optimal supervised deep-learning model. By incorporating recurrent neural nets and deep perceptron nets, our deep-learning model can capture both temporal and network effects in a customer's purchase history; further, it can capture complex customer-firm interactions by learning individual-level purchase transaction trajectories, store-store shopping networks, and other latent high-dimensional customer features. In doing so, our model addresses the common issues in most predictive models such as imbalanced training, data sparsity, temporality, and scalability. Moreover, our model can identify the heterogeneous treatment effects for various features.

We then apply the validated deep-learning model to identify target customers from the company's whole customer base and to benchmark sales effectiveness. The performance of this deep-learning based targeting is benchmarked against common industry practices, where firms target customers who are frequent shoppers or high spenders in the past, as well as traditional machine-learning approaches, where predictive models can be built to infer and select target customers. That is, we use the deep-learning model as a targeting rule to identify high-value users and improve campaign targeting effectiveness in terms of sales revenues for the company.

Our application with a large department store on a total of 2.8 million customers with billions of transactions demonstrates that our deep-learning model achieves high targeting accuracy in predicting purchase likelihood and significantly boosts sales performance relative to common industry practices and traditional machine learning approaches. In terms of the empirical mechanisms, we show that the customer targets identified by our deep-learning model differ from those identified by industry common practices and traditional machine learning approaches, and that certain customer features can account for those differences in customer selection and targeting performance. Thus, marketers may achieve sub-optimal customer targeting effectiveness not because they offer the wrong incentives, but they leverage the wrong targeting rules and select low-value customer targets.

Overall, we contribute to the literature in the following aspects: 1) we propose a framework combining field experiments and machine learning to understand the response heterogeneity to marketing campaigns; 2) we benchmark our dynamic hybrid deep learning model against several state-of-the-art baselines and show its superior performance for both in-sample and out-of-sample tests; 3) Our model captures various explicit and implicit patterns among users and stores, such as temporality and network effects.

Literature Review

There is a large body of work in the literature addressing the heterogeneous treatment effect estimation and user purchase prediction and targeting. We only overview some pertinent studies next.

Heterogeneity in customer responses to targeted campaigns. Various randomized experiments have been designed to better understand customer heterogeneity to improve targeting and user purchase

prediction. For example, Ascarza (2018) encourages firms to broaden the use of randomized experiments and finds that targeting heterogeneity in the sensitivity to retention programs is more effective than targeting customers who are at risk of churning. Dholakia (2006) conducted a field experiment with a large automobile servicing firm to understand who should be the targets with discounts offered and showed that sending a small amount of incentives to the right customers (e.g., who previously paid full price) can actually lead to less demand. Researchers have also investigated whether advertising impacts purchasing, which advertising messages are most effective, and how to design optimal ads using field experiments (Agarwal et al. 2011; Andrew et al. 2016; Anderson and Simester 2013; Dube et al. 2017; Lambrecht and Tucker 2013; Lewis and Reiley 2014; Li et al. 2017; Tucker 2014; Tucker and Zhang 2011; Yang and Ghose 2010). All of these might involve a significant manpower investment from both researchers and organizations, an unusual and difficult feat. See a comprehensive review of field experiment studies modeling heterogeneous treatment effects in Simester (2017). Foster et al. (2011) propose a random forest to estimate the effect of covariates on outcomes in treated and control groups. The difference is then linked to treatment effects of units' attributes using regression or classification trees. Liberali and Hauser (2018) used multi-armed bandit solutions to develop more efficient experiment designs with fractional observations and adjusted statistical power for the heterogeneous suboptimal treatments. Wager and Athey (2018) developed tree-based methods for estimating heterogeneous causal effects with statistical guarantees, named causal forest. Others (Imai and Ratkovic 2013; Tian et al. 2014; Weisberg and Pontes 2015) developed lasso-like statistical methods for causal inference in a sparse high-dimensional linear setting. Most of these nonparametric methods are not good at dynamic learning.

User purchase prediction in machine learning. There is a growing literature about predicting user purchasing and targeting customers with machine learning methods. Fang et al. (2013) formalize the product purchase for a user as a link prediction problem and develop a locally weighted expectation-maximization method to predict adoption probabilities. Similarly, Li et al. (2015) propose a utility-based link recommendation method based on the value, cost, and linkage likelihood. The common issue in these methods is imbalance, where the number of positive instances (e.g., links) is far less than the number of negative ones. Thus, various collaborative filtering (Melville et al. 2002; Sarwar et al. 2010), matrix factorization (Koren et al. 2009), matrix completion (Candes and Recht 2008), and other recommender system algorithms have also been used. Such matrices are usually extremely sparse and even the user-product preference data is not available. Also, researchers leverage user profile information in a social network (e.g. Facebook) to predict what categories of products the user will buy from (Zhang and Pennacchiotti 2013). For targeting, Zhang et al. (2016) designed a network-based algorithm with text mining that identifies users' interest in Facebook brands based on their and other users' historical activities for social advertising. Liu et al. (2016) claim that merchants can identify customers who can be converted to regular, loyal buyers and then target them to reduce their promotion costs and increase returns on investment. We extend prior works by developing deep-learning algorithms with causal inference to enhance the effectiveness of targeted campaigns.

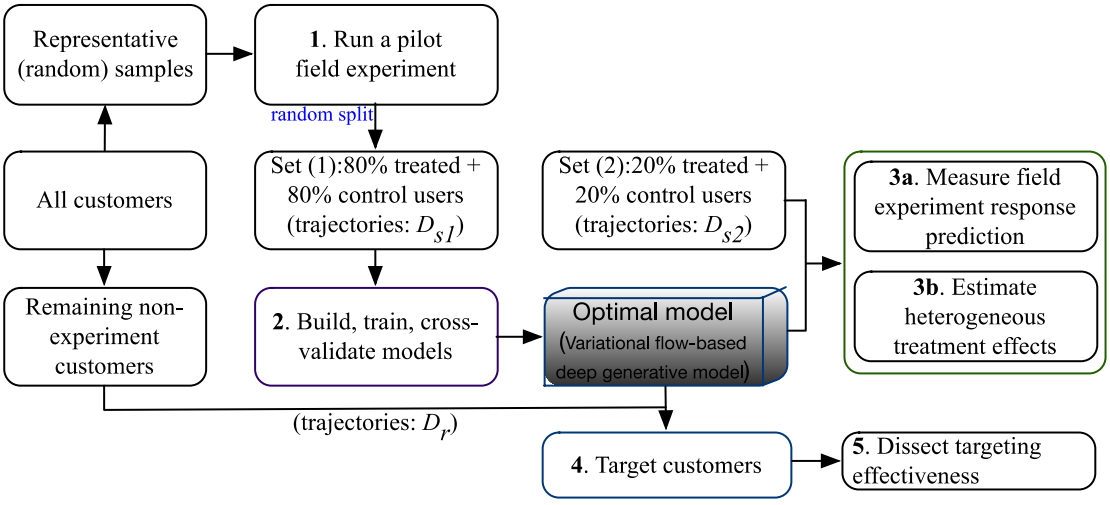


Figure 1. Our proposed framework to select high-value customer targets

Methodology

Features

In this section, we will describe the process of extracting explicit and implicit features from shopping transactions. The objective is to identify relevant features for our learning models to achieve better performance.

Features related to members alone. For each member, we have their profile information, such as their demographics (sex, age, address, and contact), and their membership type, which is based on the earning points of past shopping (different levels of membership represent different percentage of discounts for their next purchases). Some members might be frequent enough shoppers to gain sufficient points to become VIPs. We denote this set of features as fprofile. In this study, we only use gender ($f_{profile\_sex}$), age ($f_{profile\_age}$), and membership type ($f_{profile\_mtype}$), since others have many missing values.

Features related to individual stores and store-store network "relations". In our data, each store represents one brand's store inside the big department store compound. This feature allows us to build an implicit store-store network to capture relevant information (e.g., store "relations"). The intuition is that if any customers purchase from two common stores, these store brands may offer products with inherent relations (e.g., complementary brands selling related product categories like shoes and pants). Thus, such store-store network may have valuable information about purchase patterns. In this network, stores are designated as nodes, and an edge between two stores is formed if there exists common customers with activities in both stores (i.e., members who had purchases with both stores in our data collection period). The larger the number of common customers across two stores, the higher is the weight of the edge connecting the two stores. This network represents store affinity. We define an undirected and weighted store-store network (denoted as SSN) as SSN = <V, E>, where the set of nodes V corresponds to stores and the set of edges E carries weights that represent the number of common members between any two stores. Formally, $V = \{S_i\}$ with Si being a store having mi as the set of members who have purchased from this store in our study time period, and $E = \{(S_i, S_j) | S_i \cap S_j \notin \emptyset\}$ with corresponding weight $w_{ij} = |S_i \cap S_j|$. Here, $1 \leq i, j \leq N$, and N is the total number of unique stores, which is 3,478 in this study.

Normalization of the store-store network: More popular stores typically attract more members and thus have larger quantities of transactions. Such stores with a larger number of active members will then have larger numbers of common members with connected stores, compared to stores that are not as popular, which will lead to the more popular stores having much larger edge weights in the store-store network. If not accounted for, these higher weighted edges associated with a few very popular stores could dominate analyses in the network. To facilitate comparison across stores in the network, edge weights must be normalized. In this study, we use a simple two-step approach similar to Zhang et al. (2016) to normalize the network while preserving the semantic integrity of the network as much as possible. The following was the process we undertook to ensure this. Step I: we first obtain an intermediate weight of an edge connecting two stores S_i and S_j: $W'_{ij} = \frac{w_{ij}}{m_i * m_j}$, where m_i and m_j are the number of active members for store S_i and S_j, respectively, in the study time period. W_{ij} is the original weight of the edge before Step I. Step II: We then normalize all intermediate weights W'_{ij} by setting $W^*_{ij} = \frac{W'_{ij}}{max_{\forall(i,j)}\{W'_{ij}\}}$. A toy example of this normalization process is show in Figure 2.

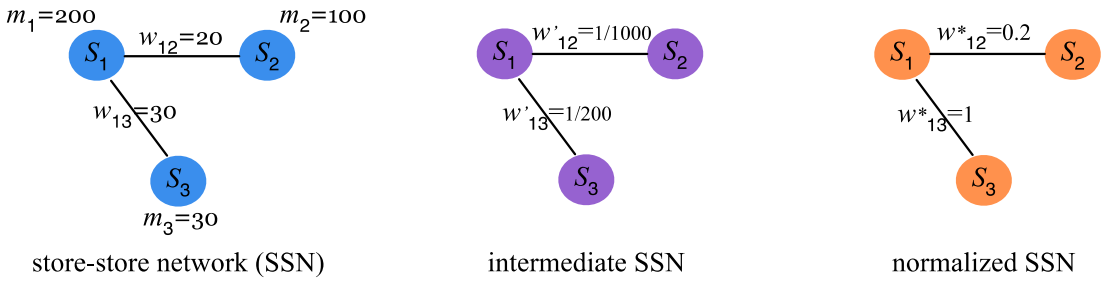


Figure 2. A toy example of the store-store network normalization

Once we have the normalized network *SSN*, we can identify several relevant network-based features (denoted by fnet). For example, the eigenvector centrality score ($f_{net\_ec\_i}$) represents how popular the i^{th} store is, whereas to reflect the connectedness (similarity to other stores) of the i^{th} store, At the user level, we take the average of fnet of all stores the user purchased as the network feature for that user, formally, the network eigenvector centrality score for the user u is defined as $f_{net\_ec\_u} = \frac{1}{n_u}\sum_{i=1}^{n_u} f_{net\_ec\_i}$, where n_u is the number of stores the user u visited. Similarly, the network weighted average degree for the user u is defined as $f_{net\_wad\_u} = \frac{1}{n_u}\sum_{i=1}^{n_u} f_{net\_wad\_i}$.

Features related to user-store interactions. Our data not only provides details for each transaction, but also the temporal information about transactions. In this paper, we separate them into history-related features (denoted by f_{hist}) and store-related features (denoted by f_{temp}). Specifically, we have the number of purchases, denoted by $f_{hist\_npurchase\_m}$ (Visiting the same store in the same day counts one purchase), the number of unique stores visited ($f_{hist\_nstores\_m}$), the number of products ($f_{hist\_nproducts\_m}$), and how much spent ($f_{hist\_cost\_m}$) for the mth month in the pre- treatment period for f_{hist} (m=1, 2, 3, and 4). As we mentioned before, we believe that the temporal information embedded in transactions (particularly a sequence of purchases from different stores) is one of effective factors for prediction. However, there is no prior work in conventional machine learning that has identified useful features that represent such temporal information. Instead of performing handcrafted feature engineering, we rely on recurrent neural network models to discover intricate patterns of user purchasing behavior using the raw ordered purchase sequence data. We denote this feature as f_{temp}. For example, f_{temp} = {($store_i$, $nproducts_i$, $cost_i$), ($store_j$, $cost_j$, $nproducts_j$), … ($store_k$, $nproducts_k$, $cost_k$)} means a user's sequence of purchases from different stores (where i, j, … k are ordered by time; $store_i$: store id; $nproducts_i$: the number products purchased from the i^{th} store; $cost_i$: how much spent from the i^{th} store). Table 2 presents the descriptive of features for experiment members in the pre-treatment period. Each user is represented by a combination of various features.

Models

Deep learning evolved from an already existing machine learning technique called the artificial neural networks (ANN) first formally invented in 1958 (Rosenblatt, 1958). The feed forward deep network or multilayer perceptron (MLP) was developed to learn mathematical functions mapping some set of input values to output values. Since the computer infrastructure (both hardware and software) has improved, especially distributed computing and stochastic gradient descent, sophisticated deep-learning models (such as convolutional neural networks, recurrent neural networks, etc.) have been achieving many successes in various areas (LeCun, 1998; Hochreiter and Schmidhuber, 1997; Liu et al., 2016). Essentially, it models high level abstractions and patterns in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations. It solves the problem in representation learning by introducing representations that are expressed in terms of other, simpler representations. Further, it enables the computer to build complex concepts/patterns out of simpler ones, involving many improvements in techniques to overcome the shortcomings found in previous artificial neural network model estimation (LeCun et al., 2015).

In this study, we utilize deep learning as supervised learning classifiers to predict whether experiment users purchase in the post-treatment period. Before introducing deep learning models, we first explain the intuition behind conventional machine learning to perform user purchase prediction. In essence, user purchase prediction is a binary classification problem, and therefore any existing supervised machine learning algorithms can be applied, such as support vector machines (SVM), random forest, decision trees, etc. However, as mentioned before, these models are not good at capturing hierarchically deep representations of data and dealing with high- dimensional temporal data. This motivates us to develop deep learning models.

Here we introduce several deep-learning models ranging from deep neural nets, convolutional neural nets, stacked variable-length recurrent neural nets, hybrid model, to our final dynamic hybrid model (see Figure 3).

Model A. The first model we implement is the deep neural nets (DNN) with 6 fully connected hidden layers with [128, 128, 256, 256, 512, 512] neuron units each (see Figure 3A). The input provided to the

DNN consists of the following features: user profile: $f_{profile}$ ($f_{profile\_sex}$, $f_{profile\_age}$, $f_{profile\_mtype}$), network characteristics: f_{net} ($f_{net\_ec\_u}$, $f_{net\_wad\_u}$), user purchase history: $fhist$ ($f_{hist\_npurchase\_m}$, $f_{hist\_nstores\_m}$, $f_{hist\_nproducts\_m}$, $f_{hist\_cost\_m}$), where m=1, 2, 3, and 4, and individual store purchase: {($f_{store\_nproducts\_1}$, $f_{store\_cost\_1}$), ($f_{store\_nproducts\_2}$, $f_{store\_cost\_2}$), …, ($f_{store\_nproducts\_n}$, $f_{store\_cost\_n}$,)}. The feature f_{temp} is not used because this model is not capable of capturing temporal patterns. *DNN* is prone to overfitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data. Several techniques have been proven to be effective for avoiding overfitting, such as *regularization* (ll_1 and ll_2) and *dropout* (randomly omits units from the hidden layers with a probability of p during training).

Model B. When dealing with multi high-dimensional inputs such as images, it is impractical to connect neurons to every neuron in the previous layers because such network architecture does not take the spatial structure of the data into account. Also, *DNN* is not scale well for high dimensional data due to the curse of dimensionality. Convolutional neural nets (ConvNets) were proposed to exploit this spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers: each neuron is connected to only a small region. Another benefit of using ConvNets is the parameter sharing to control the number of free parameters. Furthermore, ConvNets involves pooling which progressively reduces both the spatial size of the representation as well as the number of parameters and amount of computation in the network, and hence to also control overfitting. ConvNets usually involves many specific hyper-parameters, such as the number of filers (kernels), filter shape, stride, pooling shape, and padding, etc. How to choose a combination of these parameters to build a ConvNets is not easy. What researchers including us usually do is adopt and modify some existing successful models (Krizhevsky et al. 2012; Simonyan and Zisserman 2014; He et al. 2015) accordingly to fit the specific task. In this paper, we apply *VGG16* (Simonyan and Zisserman 2014). The feature variables used in this model are the same as Model A.

Model C. The three models we have built so far ignore important features f_{temp}. To have signals taking temporal information into account to learn long-term dependencies, we build a recurrent neural network (RNN). An example in Figure 4a shows different purchase sequences for two users (green dotted arrow: $S_1S_2S_5$, red dotted arrow: S_1S_3). The width of edges in the store-store network represents the correlation of two stores (e.g., selling complementary products). Since S_3 is weakly connected to other stores, users are more likely not to make next purchases after S_3 comparing to the case that a user reached S_2. Several variants of RNN were developed, and among which LSTM (long short term memory) is very standard and commonly used (Hochreiter and Schmidhuber 1997). Since the number of historical purchases is varied for different members, in this paper we stack three LSTM layers with a variable-length input (called *stacked VL-LSTM*). Each LSTM has 128 cells (neurons). To ensure to have the same length within each batch, we use zero-padding. The input features in this model are f_{temp} only. To avoid overfitting, we adopt dropout. GRU (gated recurrent unit) a simpler LSTM can be considered as an alternative.

The stacked VL-LSTM model is designed to well capture temporal information but neglect other important features such as network characteristics, history, and profile. How to build a hybrid model to combine all features becomes our focus. Figure 3B shows a hybrid model with a LSTM component to intake sequential features, a separate component to place static features, such as profile and networks, and a merging component to combine the former two, called *HybridNN*. The input is all features we identified to represent each user. We also use *dropout* to avoid overfitting.

HybridNN is still a "static" model in a sense that the model is built once based on the pre-treatment data and applied to predict user purchase likelihood regardless of the length of post-treatment. Building a real dynamic and adaptive model to improve the prediction power as time evolves becomes the focus of our final model. Specifically, every time we obtain some ground-truth purchase data in the post-treatment, we adjust the HybridNN model by reinforcing the learned patterns from the incorrectly predicted instances/members (e.g., increasing the weights proportionally to the number of instances in the training set). We name this model as *Dynamic-HybridNN* (Figure 3C).

For all models, the final output layer is the softmax layer/function, a generalization of logistic function that converts a vector of real values to a K-dimensional vector in the range of 0 and 1 that add up to 1, $\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$, for i = 1,2, ..., K and K=2 in our binary classification.

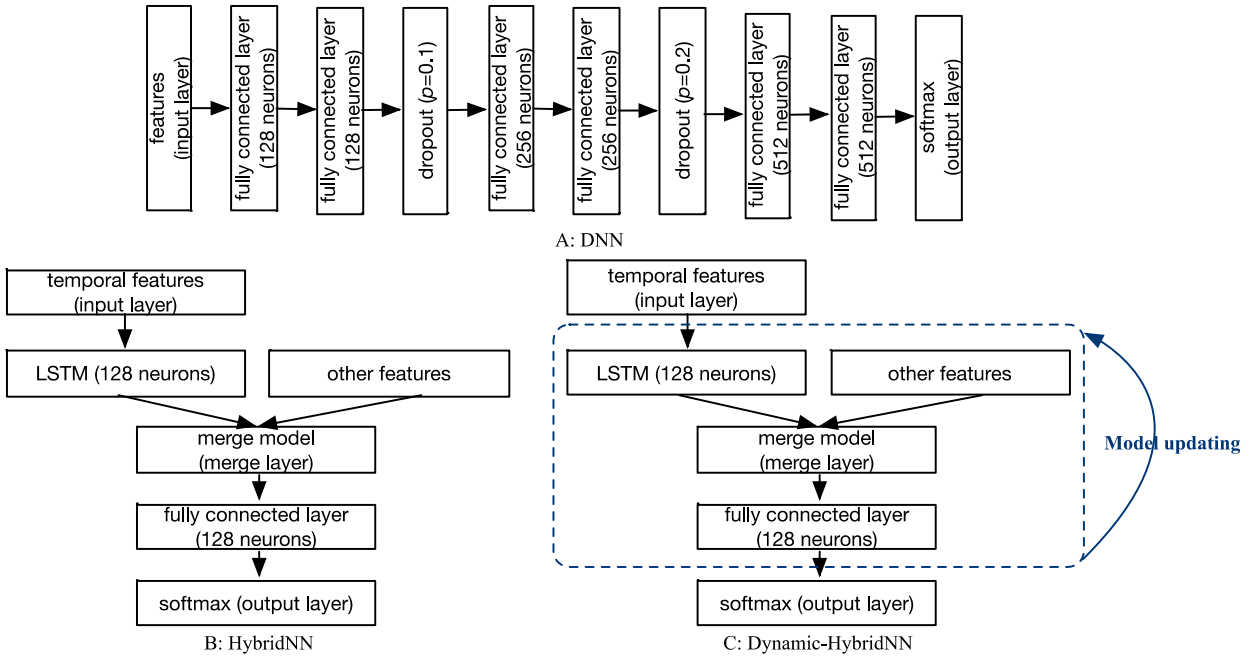


Figure 3. Architecture of different deep-learning algorithms

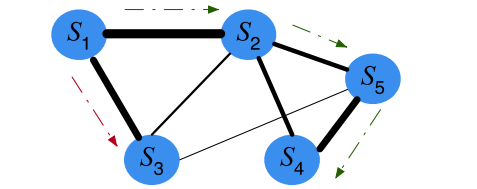


Figure 4(a): A toy example of purchase sequences.

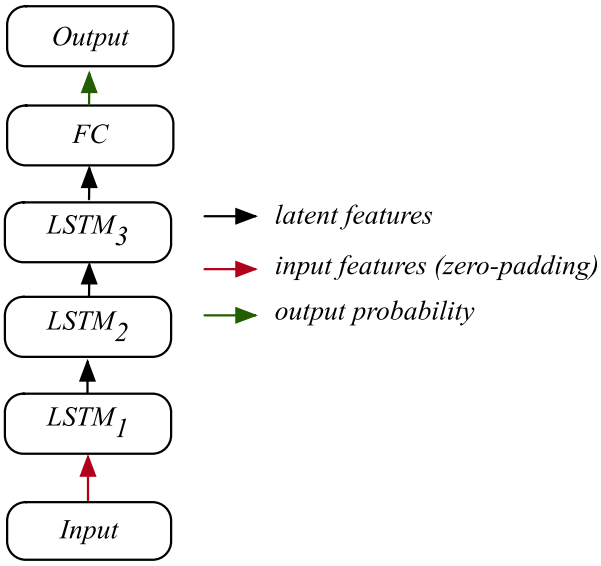


Figure 4(b): Architecture of ***VL-LSTM***.

Figure 4. VL-LSTM model capturing temporal sequence data information

Application

Our application involves a large department store (about 2.81 million customers with billions of transaction records of product purchases) in an Asian city. The department store compound has different individual stores selling many different categories of products[1], such as children's clothing, jewelry, candy, health & beauty, and others. The pilot field experiment is designed as follows. We randomly select and assign a small sample of store members (about 34,300 customers) to either the treated group (with incentives of $20 awarded for purchasing) or the control group (no such incentives). We then collect data on the shopping transactions of these members for a 4 month pre-treatment period and 2 month post-treatment period. This data consists of the transactions from experiment members and remaining non-experiment members. Among experiment members, we further randomly select 80% from the treated group and 80% from the control group to form set (1) and the remaining 20% to form set (2). Thus, we have (i) transactions for set (1) experiment members (denoted by D_{s1}); (ii) transactions for set (2) experiment members (denoted by D_{s2}); and (iii) transactions for remaining non-experiment members (denoted by D_r). Each transaction is related to one product and consists of multiple pieces of information: demographics of the purchasing member (such as cell phone, home phone, address, membership type (VIP or not), age, and gender), purchasing information (such as from which individual store, the number of product purchased, the product price), and the transaction timestamp. The descriptive statistics of our dataset are shown in Table 1 and Table 2 ('-' indicates "not applied"). Note that the randomization check we conducted is satisfied, since the *p*-values for all available user characteristics are greater than 0.5.

As shown in Figure 1, given all experiment users in set (1) and their corresponding transactions (D_{s1}) in the study period, we represent each user using features mentioned above to obtain a training set[2] $T_{training} = \{(f_1^i, f_2^i, \cdots, f_k^i, l^i) \mid 1 \le i \le N_{train}\}$ where N_{train} is the total number of unique experiment members in set (1); k is the size of the features; l^i is the binary label for the i^{th} member (1 if the member purchases in the post-treatment period, 0 otherwise); Then we build, train different models (M_1, M_2, ..., M_n) and select an optimal one (M^*) using in-sample testing (D_{s1}) and out-of-sample (holdout) testing (D_{s2}) based on the metrics defined in the results section. Finally, we use the M^* and D_{s2} to estimate heterogeneous treatment effects of various features to help understand customer heterogeneity for better targeting.

	Pre-treatment	Post-treatment
# of total transactions	1,601,964	602,189
	Treated	Control
# of experiment users	9,009	25,316

Table 1: Descriptive statistics of dataset

	Feature (*f*)	**MIN**	**MAX**	**AVG**	**STD**
$f_{profile}$	$f_{profile\_mtype}$ membership type	-	-	-	-
	$f_{profile\_sex}$ gender	-	-	-	-
	$f_{profile\_age}$ age	24	63	30.35	9.13

[1] Individual store means one brand-name store inside the big department store compound. For the rest of the paper, we mean individual stores when referring to stores unless explicitly indicated.

[2] We also include the data from the first week of post-treated period into the training mainly because this is used to adjust weights for dynamic hybrid model.

f_{hist}	$f_{hist\_nstores}$ # unique stores	0	22	8.237	2.67
	$f_{hist\_npurchase}$ # visits	0	52	10.146	3.87
	$f_{hist\_nproducts}$ # products	0	230	13.733	8.54
	f_{hist_cost} $ cost	0	32900	668.79	648.59
f_{net}	f_{net_ec} eigenvector centrality	0.0074	1.0	0.637	0.142
	f_{net_wad} weighted avg degree	0.0016	0.748	0.116	0.121

Table 2: Feature descriptive of members for the pre-treatment period

Results

Evaluating and Validating Deep-Learning Models with Field Experiment Data

Evaluation metrics. We evaluate the performance of different machine learning models from (1) in-sample model fitting, and (2) out-of-sample (holdout) model prediction. To measure model fitting, we introduce several standard metrics, such as precision (*P*), recall (*R*), and F-measure (*F*). Note that the higher their values, the better the performance. Precision is the accuracy over the cases predicted to be positive. Recall is the same as true positive rate. F-measure is the harmonic mean of precision and recall at a given point. Specifically, they are defined as follows.

$$\text{Precision } (P) = \frac{TP}{TP+FP}$$

$$\text{Recall } (R) = \frac{TP}{TP+FN}$$

$$F\text{-measure } (F) = 2 \cdot \frac{precision \cdot recall}{precision+recall} = \frac{2TP}{2TP+FP+FN},$$

where TP is the quantity of true positives/purchases (i.e. the number of positive cases correctly classified into the positive class), FN is false negatives (i.e. the number of positive cases incorrectly classified into the negative class), and FP is false positives (i.e. the number of negative cases incorrectly classified into the positive class). All results reported in this study are 10-fold cross validation-based to avoid overfitting. For the out-of-sample (holdout) model prediction, we compare the predicted response with the ground true response of the subset users (D_{s2}).

Baselines. To select an optimal machine learning model, we compare with several well-known baselines (SVM and Random Forest) that perform well in general. SVM can efficiently perform a non-linear classification by applying the kernel trick to maximum-margin hyper-planes, implicitly mapping inputs into high-dimensional feature spaces. Some common kernels include linear, nonlinear, polynomial, Gaussian radial basis function (RBF), and Hyperbolic tangent (Cortes and Vapnik, 1995). In this study, we choose RBF. Likewise, Random Forest is an ensemble learning method that performs classification by constructing a multitude of decision trees at training and outputting the class that is the mode of classes of the individual trees (Tin Kam 1995). It is a way of averaging multiple deep decision trees, trained on different parts (either feature or data sample) of the same training set, with the goal of reducing the variance. This comes at the cost of a small increase in bias and some loss of interpretability, but generally boosts the prediction performance (Hastie et al. 2002).

In-sample model fitting. We train various models, plus two baselines, on D_{s1}. Figure 5 shows that *Dynamic-HybridNN* predicts user purchasing very well in *F*-measure when compared to other models and baselines (also performs the best on other metrics. The results can be provided upon request). Note that *Y*-axis is the *F*-measure. Since the training set is highly imbalanced, we use an over-sampling technique (e.g., SMOTE) where we randomly select approximately the same amount of negative customers as positive customers. The results demonstrate that it is fairly consistent across the three different training sets (different randomly selected negative customers). The temporal pattern in purchase

history has a bigger impact on the overall performance over the spatial structure, which explains why the ConvNets model does not outperform DNN, and models with the LSTM component generally work well.

Note that some common hyper-parameters used in this paper to report results are: the number of epochs: 10; the batch size: 256; the validation size: 10%; activation function: ReLU; optimizer: Adam; loss function: binary cross entropy. We also empirically tried different settings regarding the number of neurons and the number of layers used in DNN and reported the one with the best performance.

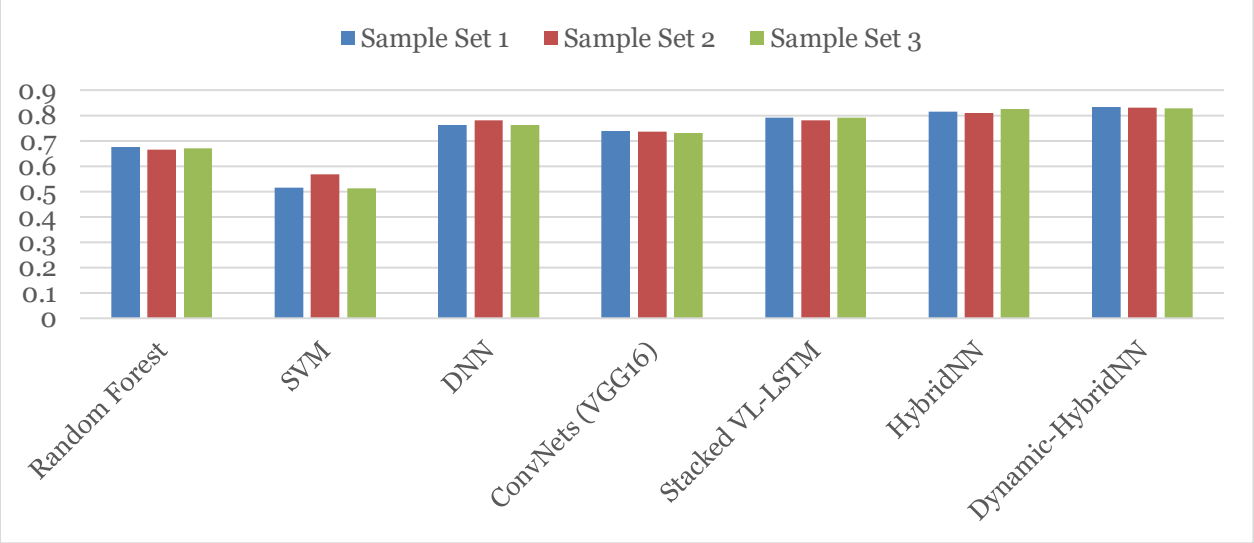


Figure 5. Model comparisons

Out-of-sample (holdout) model prediction. We apply *Dynamic-HybridNN* to predict the ground truth for the experiment users in the set (2) (we have ground truth, or the real purchase records, after the campaign for set (2) because they are part of the pilot field experiment). The purchase rates from the ground truth and our *Dynamic-HybridNN* prediction are computed, as shown in Figure 6. Specifically, (i) the predicted purchase rate is fairly close to the ground truth. The ground truth for the purchase rate of the treated is .221 and the *HybridNN* prediction is about .182, while the ground truth for the purchase rate of the control is .156 and the prediction is about .121. And (ii) the model has a higher prediction power for the treated users than users in the control group, which demonstrates that our deep-learning model can indeed capture the effect of marketing campaigns on average. We make predictions using 6 other models and all obtain inferior results to *Dynamic-HybridNN* for users in both treated and control groups. Note that there is a significant treatment effect of the $20 incentives, because the purchase rate of the treated group is statistically significantly higher than that of the control group ($p < .01$). This confirms that monetary incentives can boost customer purchases on average. There is a tradeoff to be made when training a supervised model. With larger training sample, the model is better at capturing informative patterns, but it requires more computational cost. Figure 7 plots the *F*-score of the *Dynamic_HybridNN* as a function of the size of the training sample. It suggests that performance stabilizes after 50% out of total training samples (about 17,000+ sequences).

Detecting Heterogeneous Treatment Effects

To estimate treatment effect for a given feature (f), we first apply the optimal model to transactions of the 20% treated users (in D_{s2}) to predict their purchase probabilities. Then the difference of overall predicted purchase rates between including feature f and excluding feature f is calculated, denoted by $\Delta_{treated}=prate_F - prate_{F-f}$, where F is the set of all features. Similarly, we obtain $\Delta_{control}=prate_F - prate_{F-f}$ for the control group. Then the treatment effect of feature f is $TE_f = \Delta\Delta_f = \Delta_{treated} - \Delta_{control}$.

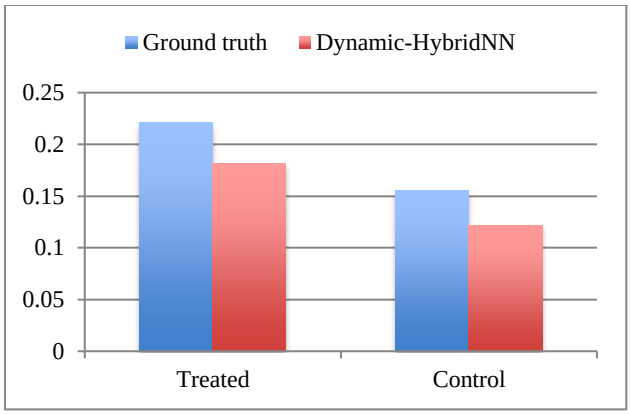


Figure 6. Ground truth vs. predicted purchase rate for both treated users and control users in the D_{s2} of the pilot field experiment

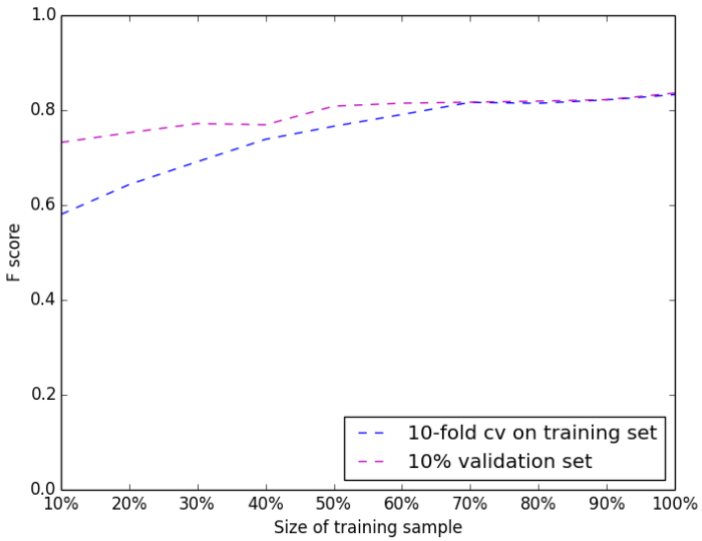


Figure 7. F-score as a function of the size of the training sample

Table 2 presents the heterogeneous treatment effects of various features. It shows that temporal information (f_{temp}) in purchase transactions has the largest effect. For profile, age has a two times larger effect over membership type. But the overall effect is small, which means that demographic information is not the key factor driving members to purchase under that marketing campaign. All history related features have similar and large effects. The key here is to identify high-value customer targets, as discussed next.

	Feature (f)	$\Delta_{treated}$	$\Delta_{control}$	$\Delta\Delta_f$	***p*-value**
$f_{profile}$	$f_{profile\_mtype}$ membership type	0.01768	0.01369	0.00399	0.232
	$f_{profile\_sex}$ gender	0.01645	0.01520	0.00125	0.788
	$f_{profile\_age}$ age	0.01468	0.00686	0.00782	0.004
f_{hist}	$f_{hist\_nstores}$ # unique stores	0.03308	0.00289	0.03019	< 0.001

	$f_{hist\_npurchase}$ # store visits	0.05123	0.01776	0.03347	< 0.001
	$f_{hist\_nproducts}$ # products	0.03661	0.00665	0.02996	< 0.001
	f_{hist_cost} $ cost	0.05880	0.02479	**0.03401**	< 0.001
f_{net}	f_{net_ec} eigenvector centrality	0.03560	0.00038	0.03022	< 0.001
	f_{net_wad} weighted avg degree	0.02703	0.01263	0.01440	< 0.001

Table 2: Heterogeneous treatment effects of features

Identifying Customer Targets to Enhance Targeting Effectiveness

We finally apply the learned deep-learning model to the company's remaining customer base (D_r) to identify target customers and to benchmark the sales effectiveness. To select customers to target, companies should first prioritize customers with the highest purchase likelihood, as this will increase the effectiveness of the campaign (e.g., incentive). Second, the value of the predicted purchase likelihood should be used not only as a "ranking" metric to better allocate resources, but also as a method to determine which customers should be targeted – that is, who should receive the incentives. We compare this deep learning targeting with industry common practice (e.g., past purchase frequency or spending amount), as well as baseline approaches regarding the targeting accuracy in predicting the purchase likelihood and sales performance.

In terms of recovering empirical mechanisms for the results, we show that the customer targets identified by this deep-learning model may not be fully overlapped with those identified by industry common practice and baseline models. Further, we also examine how some customer features can account for this difference in customer target selection and targeting performance effectiveness.

The impact of marketing campaign if targeting based on Dynamic-HybridNN. We now compare the impact of the marketing campaign if the firm targets the same proportion of customers but with different targeting rules. We consider four targeting rules: targeting based on our optimal deep-learning model *Dynamic-HybridNN* vis-à-vis the two common industry practices: high visit frequency (denoted B1), and high spending amount (denoted by B2), as well as one traditional machine learning method: Random Forest. In other words, we calculate the sales performance, net incentive costs, of the same campaign and same proportion of customers but with different customer targets identified by the four targeting rules. We do so with the remaining 2.776 million customers of the company. Figure 8 depicts the impact of the campaign in terms of average sale amounts under each of the targeting scenarios, assuming the company targets 10% of customers, 20% of customers, etc.

There are several noticeable patterns. First, the impact of targeting customers based on all four strategies decreases as the percentage of customers being targeted increases. This is expected, since all strategies select the "best" customers first. Second, our targeting rule based on *Dynamic-HybridNN* is substantially more effective than the three baselines. In other words, it identifies more "correct" (high-value) customers who spend more in the future, which correspondingly brings more revenue for stores/firms. Third, when the number of customers/members being targeted is small, the baseline B2 and Random Forest are close to *Dynamic-HybridNN* for sales amount. However, the users selected from B2 and Random Forest vs. *Dynamic-HybridNN* are very different, even if both reach similar average sales amount, as elaborated next.

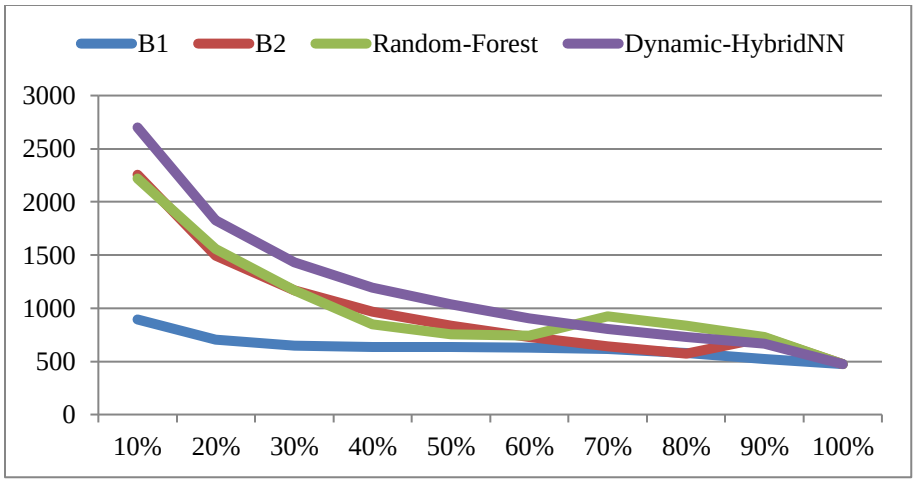


Figure 8. Different targeting effectiveness

To explain differences of the targeting rules, we quantify the overlap of targeted customers selected between the baseline (B1, B2, and Random-Forest, respectively) and *Dynamic-HybridNN*, i.e., how different the customer targets are across the targeting rules (see in Figure 9 which depicts user overlap between our targeting decision (*Dynamic-HybridNN*) and B1, B2, Random-Forest. Note that X/Y-axis is select targeting customers on TOP # decile). For each subgroup of selected target customers (e.g., top 10%, top 20%, etc.), each value point represents the percentage of customers in each top *Dynamic-HybridNN* percentile who also belong to the top B1, B2, or Random-Forest percentile. 100% means a perfect overlap between two groups (in the 45 degree line). We obtain the similar findings that we should expect to see low levels of overlap between baseline B1 and *Dynamic-HybridNN* for low top percentiles (e.g., 10%, 20%, 30%), so the selected customers are very different.

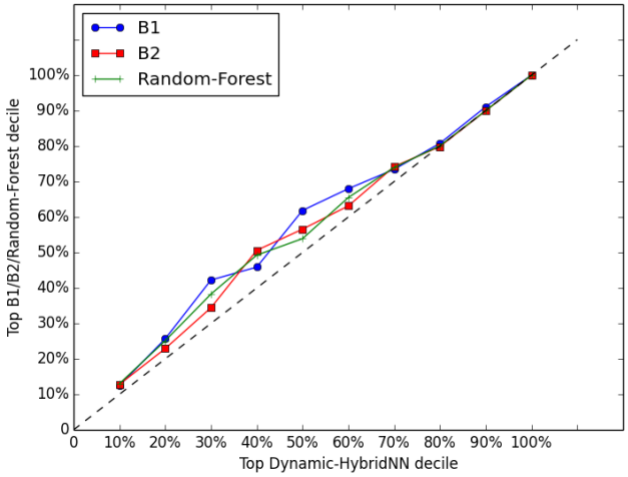


Figure 9. Different user targets selected

Further, we examine difference in the characteristics (identified features) of the targeted customers selected based on B1, B2, Random-Forest, and *Dynamic-HybridNN*. We report three representative features with large treatment effects: average age (profile, see Figure 10A. Note: average value of individual feature for selected top percentile customers by B1, B2, Random-Forest, and Dynamic-HybridNN. X-axis: select targeting customers on TOP # decile), average spending amount (history, see

Figure 10B), and average eigenvector centrality which is extracted from the store-store network as reported in online appendix B (see Figure 10C). Thus, with a low percentile of top selected customers, B1 tends to select customers who purchase more from well- known and influential stores (eigenvector centrality is large and products are relatively cheaper) and B2 tends to target those who go to high-end stores (eigenvector centrality is low and products are typically expensive), while *Dynamic-HybridNN* is in the middle.

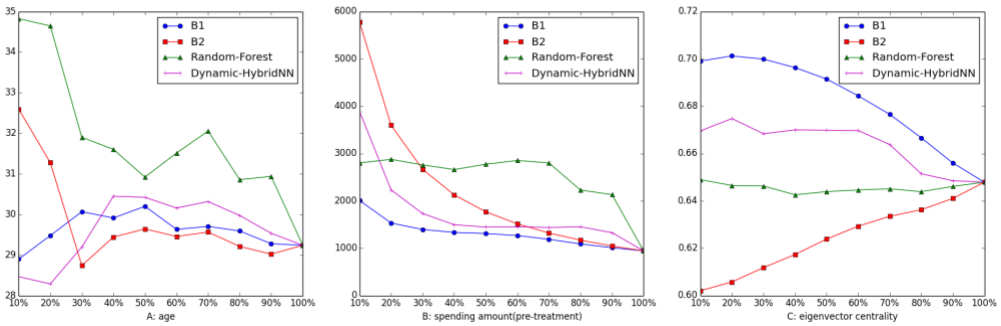
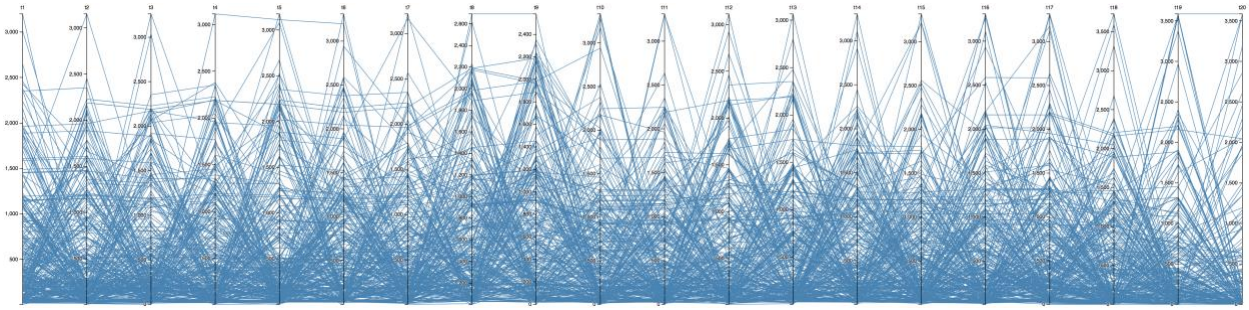


Figure 10. Different features in targeting rules

Finally, we examine the difference in temporal characteristics (ordered purchase sequence) of the targeted customers based on different targeting rules. Figure 11 shows purchase patterns of top 10% decile selected customers with the length of purchase sequence at most 20 (the average is 20.1). It is the pattern of store visit sequences for selected top percentile customers by four targeting rules. X-axis is time sequence; Y-axis is store ID (1-3748). 0 means no visits; Blue, Red, Green, and Pink represent B1, B2, Random-Forest, and *Dynamic-HybridNN*, respectively. We right pad the sequence when less than 20. From the figure, we find that: (1) sequences from users selected based on B1 are more random and less repeated; (2) sequences from users selected based on our *Dynamic-HybridNN* show an unique pattern that users are more likely to make repeated purchase from the same store (parallel connections) than others, i.e., capturing more time dependence information in the sequence; (3) sequences of B2 and Random-Forest are more similar to *Dynamic-HybridNN* than B1, especially for top percentiles. Purchase sequence patterns for top 20% deciles also show consistent patterns (ignored due to space limitation, but can be provided upon request).



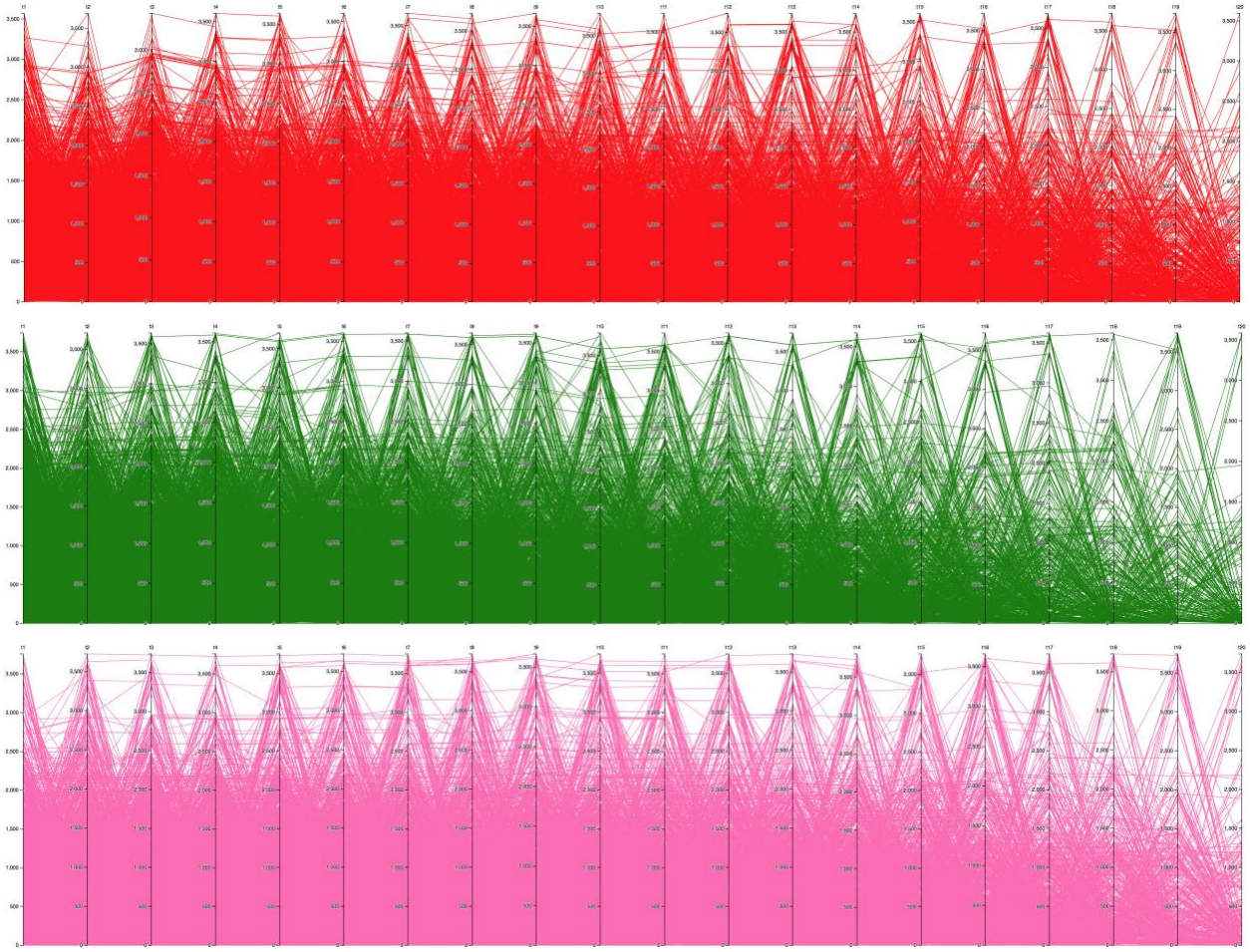


Figure 11. Different degrees of targeting temporal sequence information

Overall, these results reveal critical targeting effectiveness differences and attribute those differences to different customer targets (how many percentages of overlapped customers across the three targeting rules) and various individual feature variables. These results suggest that our deep-learning based targeting rule can lead to superior sales performance than the more common industry practices of targeting customers by past purchase frequency or spending amount. Thus, sometimes marketers may fail to achieve optimal customer targeting effectiveness not because they offer the wrong incentives to customers, but because they adopt the wrong targeting rules and sub-optimally select low-value customer targets for their campaigns.

Conclusion

This paper integrates deep-learning algorithms, big data analytics, and field experiment response heterogeneity to enhance campaign targeting effectiveness. We recommend firms run a pilot randomized experiment and use the data to train various deep-learning models. We then apply the learned model to identify target customers from the remaining customers with the highest predicted purchase probabilities. Our deep-learning models can be generalizable to most business settings, as long as customer heterogeneity data are available and finding right customers to target is pivotal to marketing campaigns. Our study informs managers that beyond gauging the causal impact of marketing interventions, big data analytics, field experiments, and deep learning can be combined to identify high-value customer targets. It is a small, but novel step for the purpose of enhancing the targeting effectiveness of marketing campaigns.

References

Agarwal, Ashish, Kartik Hosanagar, Michael D. Smith (2011). Location, location, location: an analysis of profitability of position in online advertising markets. *Journal of Marketing Research*, 48(6):1057-1073.

Anderson, Eric T., Duncan Simester (2013). Advertising in a competitive market: the role of product standards, customer learning, and switching costs. *Journal of Marketing Research*: 50(4): 489-504.

Ascarza, Eva (2018). Retention Futility: Targeting High Risk Customers Might Be Ineffective. *Journal of Marketing Research* In-Press.

Candes, Emmanuel J., Benjamin Recht (2008). Exact matrix completion via convex optimization. *arXiv:0805.4471 [cs.IT]*.

Cortes, Corinna, Vladimir Vapnik (1995). Support-vector networks. *Machine Learning*, 20(3):273-297.

Dholakia, Utpal M. (2006). How customer self-determination influences relational marketing outcomes: evidence from longitudinal field studies. *Journal of Marketing Research*, 43(1):109-120.

Dube JP, Z. Fang, N Fong, X Luo (2017), "Competitive Price Targeting with Smartphone Coupons," Marketing Science, 36(6), 944-975.

Fang, Xiao, Paul Jen-Hwa Hu, Zhepeng Li, Tsai Weiyu (2013). Predicting adoption probabilities in social networks. *Information Systems Research*. 24(1):128-145.

Forbes (2015), As Brands Turn to Digital Advertising to Reach the Right Audience, Focus on Validation Is, *Forbes Insights*, 1-2.

Foster, Jared C., Jeremy MG Taylor, and Stephen J Ruberg (2011). Subgroup identification from randomized clinical trial data. *Statistics in Medicine*, 30(24):2867–2880.

Hastie, H.W., Poesio, M., Isard, S. Automatically predicting dialogue structure using prosodic features. Speech Commun. 36(1), 63-79 (2002).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, Jian Sun (2015). Deep residual learning for image recognition. *arXiv:1512.03385v1 [cs.CV]*.

Hochreiter, Sepp, Jurgen Schmidhuber (1997). Long short-term memory. *Neural Computation*, 9(8):1735-1780.

Imai, Kosuke, Marc Ratkovic (2013). Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics*, 7(1):443–470.

Koh, Pang Wei, Percy Liang (2017). Understanding black-box predictions via influence functions. *arXiv:1703.04730*.

Koren, Yehuda, Robert Bell, Chris Volinsky (2009). Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30-37.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). ImageNet classification with deep convolutional neural networks. *The 25th International Conference on Neural Information Processing Systems*, Volume 1, Curran Associates Inc., USA, 1097-1105.

Lambrecht, Anja, Catherine Tucker (2013). When does retargeting work? information specificity in online advertising. *Journal of Marketing Research*, 50(5): 561-576.

LeCun, Yann, Leon Bottou, Yoshua Bengio, Patrick Haffner (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

LeCun, Yann, Yoshua Bengio, Geoffrey Hinton (2015). Deep-learning. Nature 521, 436-444.

Lewis, Randall A., David H. Reiley (2014). Online ads and offline sales: measuring the effect of retail advertising via a controlled experiment on Yahoo!. *Quantitative Marketing and Economics*, 12(3):235-266.

Li, C, X Luo, C Zhang, and X Wang (2017), "Sunny, Rainy, and Cloudy with a Chance of Mobile Promotion Effectiveness," Marketing Science, 36 (5), September, 762–779.

Li, Zhepeng, Xiao Fang, Xue Bai, Olivia R. Sheng (2015). Utility-based link recommendation for online social networks. *Management Science*, 63(6):1938-1952. Liberali, Gui and John Hauser (2018), "Morphing Randomized Controlled Trials," Marketing Science, forthcoming. 17.

Liu, Guimei, Tam T. Nguyen, Gang Zhao, Wei Zha, Jianbo Yang, Jianneng Cao, Min Wu, Peilin Zhao, Wei Chen (2016). Repeat buyer prediction for E-Commerce. *The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, New York, NY, USA, 155-164.

Liu, Xiao, Param Vir Singh, Kannan Srinivasan (2016). A structured analysis of unstructured big data by leveraging cloud computing. *Marketing Science*, 35(3):363-388.

Melville, Prem, Raymod J. Mooney and Ramadass Nagarajan (2002). Content-boosted collaborative

filtering for improved recommendations. *The 18th National Conference on Artificial Intelligence*, Alberta, Canada,187-192.
Rosenblatt, Frank (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65-386.
Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl (2001). Item-based collaborative filtering recommendation algorithms. *The 10th international conference on World Wide Web (WWW)*, ACM, New York, NY, USA, 285-295.
Simester, Duncan (2017). Chapter 11 - Field experiments in marketing. *Handbook of Economic Field Experiments*, (1):465-497.
Simonyan, Karen, Andrew Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556 [cs.CV]*.
Tian, Lu, Ash A Alizadeh, Andrew J Gentles, and Robert Tibshirani (2014). A simple method for estimating interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association*, 109(508):1517–1532.
Tucker, Catherine E. (2014). Social networks, personalized advertising, and privacy controls. *Journal of Marketing Research*, 51(5):546-562.
Tucker, Catherine E., Juanjuan Zhang (2011). How does popularity information affect choices? a field experiment. *Management Science*, 57(5):828-842.
Wager, Stefan, Susan Athey (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*. (forthcoming).
Weisberg, Herbert I., Victor P Pontes (2015). Post hoc subgroups in clinical trials: anathema or analytics? *Clinical Trials*, 12(4):357–364.
Yang, Sha, Anindya Ghose (2010). Analyzing the relationship between organic and sponsored search advertising: positive, negative, or zero interdependence?. *Marketing Science*, 29(4):602-623.
Zhang, Kunpeng, Siddhartha Bhattacharyya, Sudha Ram (2016). Large-scale network analysis for online social brand advertising. *MIS Quarterly*, 40(4): 849-868.
Zhang, Yongzheng, Marco Pennacchiotti (2013). Predicting purchase behaviors from social media. *The 22nd international conference on World Wide Web (WWW)*. ACM, New York, NY, USA, 1521-1532.